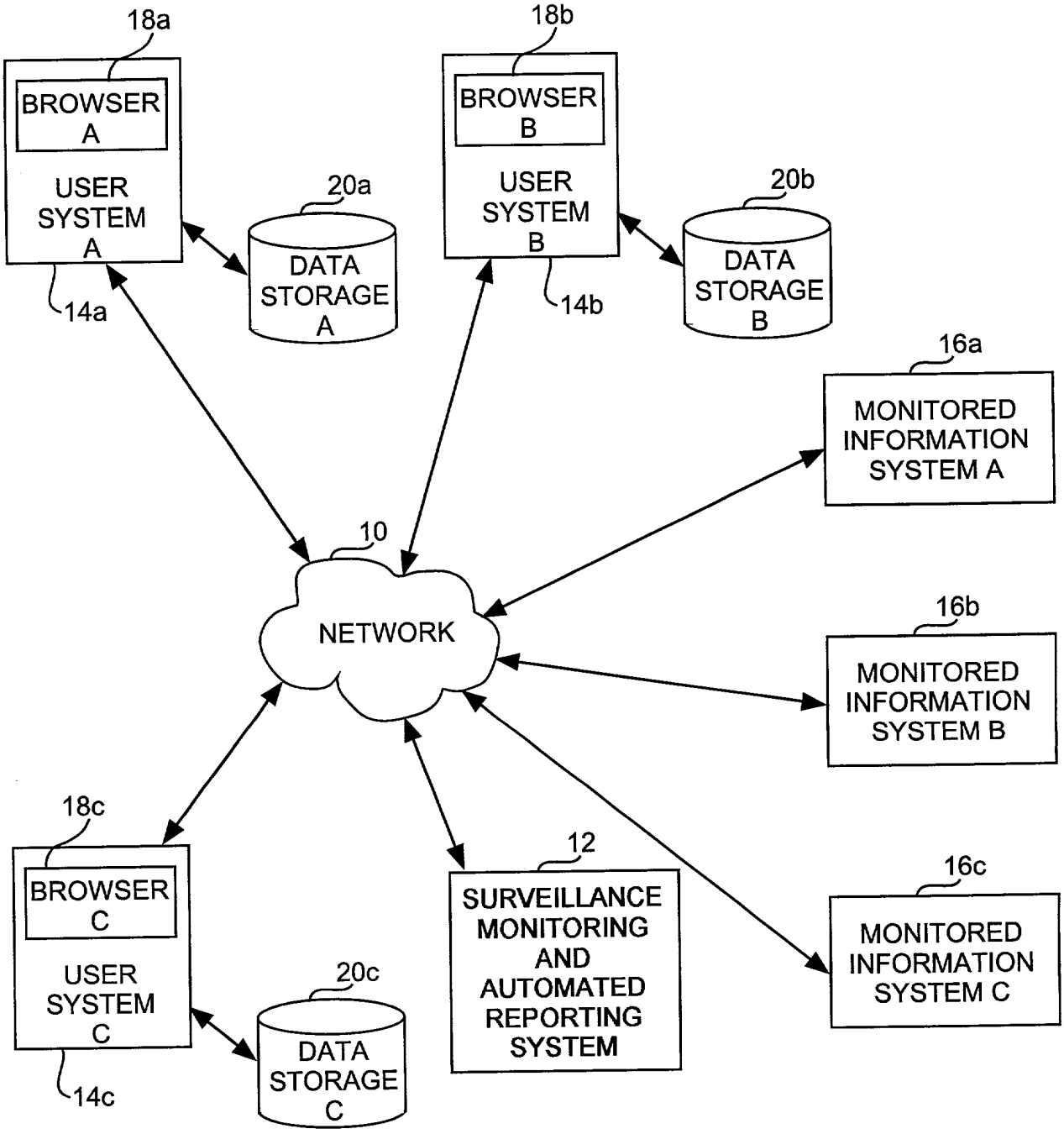
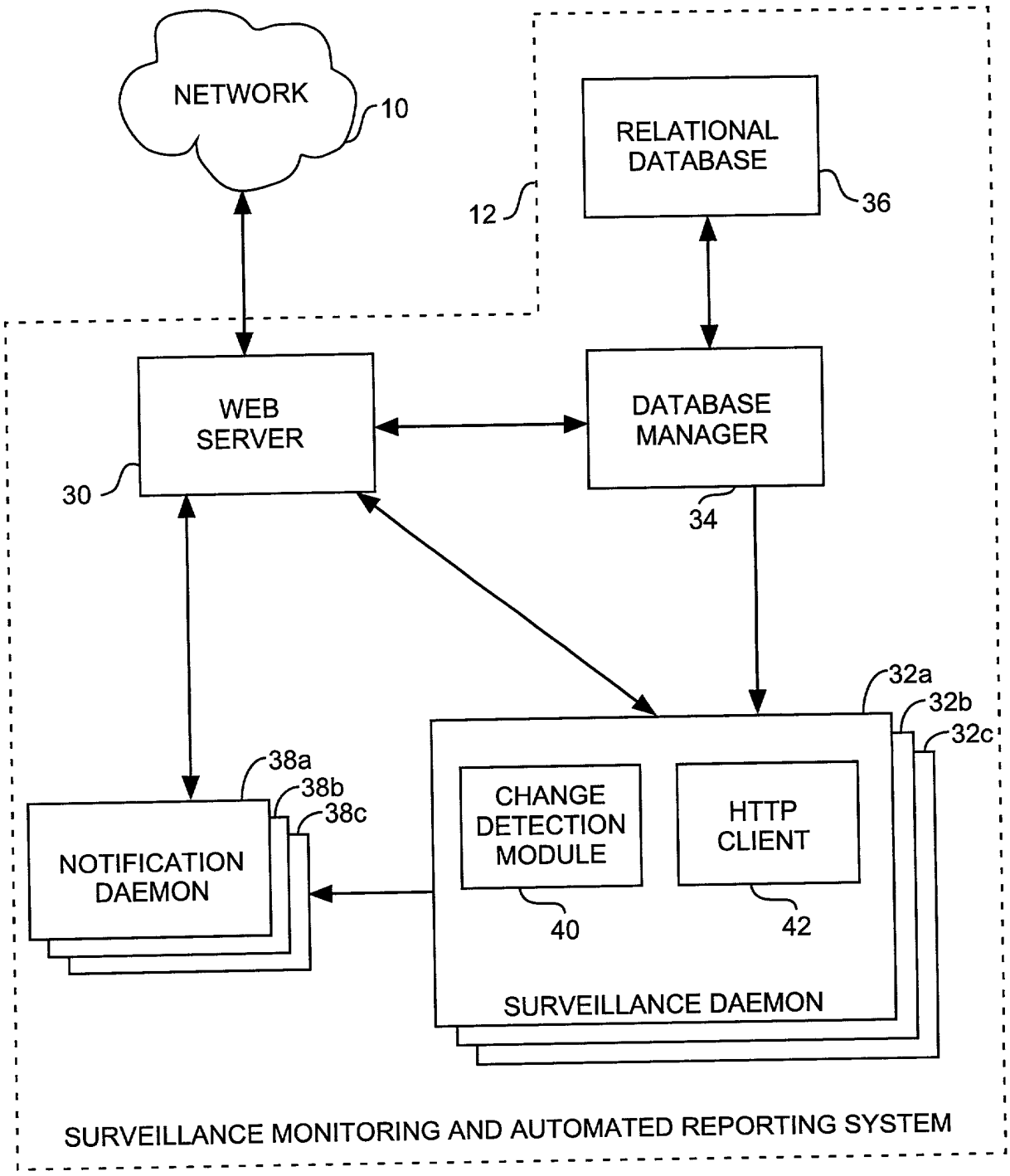


MONITORED DISTRIBUTED NETWORK

FIG. 1



NETWORK CONNECTED MONITORING AND REPORTING SYSTEM

FIG. 2

```
101   TOP LEVEL:
102
103   BEGIN
104
105   Update database: set the run flag of this daemon to TRUE;
106   Create an empty list V for link storage;
107   Set boolean Flag to TRUE;
108   While Flag is TRUE:
109   {
110         Get the user's URL, search interval SI, and depth Depth;
111
112         Do searchURL(URL, Depth, 0);
113         Get the value of the start-stop flag for this daemon from
114            the database and assign it to Flag;
115         If Flag is TRUE:
116         {
117               Sleep for SI * 60 seconds;
118               Get the value of the start-stop flag for this daemon
119                  and assign it to Flag;
120         }
121   }
122   Update database: set the run flag of this daemon to FALSE;
123
124   END TOP LEVEL
125
126
127
```

SURVEILLANCE DAEMON

FIG. 3

```
128   SUBROUTINE searchURL(URL, Depth, Level)
129
130   BEGIN
131
132   Create an empty list L for link storage;
133   Open two files for writing, an HTML file, and a formatted
134     text file (FT file);
135   Create a socket connection to the URL;
136   Request the HTML from the URL, one character at a time;
137   While there is still more data to read from the socket:
138   {
139         For each character returned from the URL:
140         {
141               Write the character to the HTML file;
142               If the character is the beginning of an HTML tag:
143               {
144                     Read all characters to the end of the current
145                       tag and store them in a string Tag;
146                     Write Tag to the HTML file;
147                     If Tag is a link to another URL:
148                     {
149                           Add the link to list L;
150                     }
151               }
152               Else
153               {
154
155                     Write the character to the FT file;
156               }
157         }
158   }
159
160   Convert the text in the FT file to one string FS, with each segment of
161     white space in FT replaced by a single blank in FS;
162
```

SURVEILLANCE DAEMON SUBROUTINE
HTTP CLIENT DATA RETRIEVAL PORTION

FIG. 4A

```
If the URL is the top-level URL (i.e., Level = 0)

{
        Do changeDetection(URL, FS);



        If there is a true change in the text:
        {
                Set TrueChange to TRUE;
                Add the keyword hit counts generated by
                   changeDetection() to the database;
                Generate an activity report;
                Add the report to the database;
                Send the report to the user via the user-specified
                   notification methods (email, pager, and/or PDA);
        }
        Else
        {
                Set TrueChange to FALSE;
        }
}

Else
{
        From the database, get the keywords associated with the
           original top-level URL;
        For each keyword W:
        {
                Count the number of occurrences N of W in FS;
                Add N to the total keyword count T;
        }
        Insert the crawled-to URL, the keyword counts N for each
           keyword W, and T into the database;
}
```

SURVEILLANCE DAEMON SUBROUTINE
CHANGE DETECTION PORTION

FIG. 4B

```
200   If TrueChange is TRUE OR Level is greater than 0:
201
202   {
203         While List L is not empty:
204         {
205               Remove the first URL link U1 from list L;
206               If Level is less than Depth:
207
208               {
209                     If U1 is not in the list V:
210
211                     {
212                           If the domain of U1 is identical to the
213                           domain of the original top-level URL:
214                           {
215                                 Insert U1 at the end of list V;
216
217
218
219                                 Do searchURL(U1,Depth,Level+1);
220                           }
221                     }
222               }
223         }
224   }
225
226   END SUBROUTINE
```

SURVEILLANCE DAEMON SUBROUTINE
RECURSION PORTION

FIG. 4C

```
301   SUBROUTINE changeDetection(URL, FS)
302
303
304   BEGIN
305
306   Retreive the formatted text P of the previous version of the URL from
307      the database;
308   If at least one character of FS is different from P:
309   {
310         Replace P in the database with FS;
311         Retrieve the boolean keyword expression Exp associated with
312            URL from the database;
313         Search the new text FS using Exp;
314         If Exp is TRUE in FS:
315         {
316               Get the keywords associated with URL from the database;
317               For each keyword W:
318               {
319                     Count the number of occurrences of W in FS;
320               }
321               Retrieve the keyword counts associated with the
322                  previous version P from the database;
323
324               If at least one keyword count for FS is different from
325                  the corresponding keyword count for the previous
326                  version P:
327               {
328                     A change between pervious version and new
329                        version has been detected;
330               }
331               Else
332               {
333                     No change is detected;
334               }
335         Else
336         {
337               No change is detected;
338         }
339   Else
340   {
341         No change is detected;
342   }
343
344   END SUBROUTINE
```

CHANGE DETECTION SUBROUTINE

FIG. 5